Android Studio 3.2.1

Instalação e Primeiros Passos

Primeira edição, 2020
ISBN 0-5487263-1-5

www.titopetri.com.br

Android Studio

O Android Studio é uma ferramenta criada pelo Google, mesma empresa responsável pela criação e desenvolvimento do sistema Android.

Ele nos possibilita criar aplicações para qualquer tipo de dispositivo que funcione com o sistema Android como **Celulares**, **Tablets**, **Relógios** (*AndroidWear*) e até mesmo o **Google Glass** (*SmartGlass*) e os futuros **Automóveis Auto-Dirigidos** (*Google Self-Driving Car*).

No Android Studio, utiliza-se a Linguagem Java para programar as aplicações e agora podemos acessar e controlar todos os recursos dos dispositivos Android para utilizar nas nossas programações:

- Botões
- Imagens
- Animação
- Música
- Internet
- Banco de Dados
- Câmera
- GeoLocalização
- Acelerômetro
- Toques e Gestos

Para programar aplicativos é indispensável conhecer todo este caminho até aqui, saber o BÊ-A-BÁ da programação.

Tudo o que aprendemos neste Livro será utilizado 100% do tempo enquanto você estiver trabalhando com o Android Studio ou qualquer outro IDE que construa softwares ou aplicativos para celulares e computadores.

Recentemente, o Google integrou uma nova linguagem ao Android Studio, o **Kotlin**. Uma linguagem com sintaxe mais simplificada e resumida que o Java.

No entanto, ainda aconselho aos alunos a aprender muito bem o Java antes de partir para qualquer outra linguagem.

Ainda existem milhares de projetos no mundo, desenvolvidos em Java.

A linguagem Java é muito utilizada ainda, e mesmo que ele venha a não existir mais algum dia, esta transição ainda demoraria muitos anos para acontecer totalmente.

Fora que o Java tem tanta regra e especificação, que no final, quem aprende o Java corretamente vai entender qualquer outra linguagem de programação (**Kotlin**, **Swift**, **Python** ou **C#**) com muito mais facilidade.

O Java possui regras e boas práticas que são comuns a qualquer linguagem.

Se o programador entende bem de Java, qualquer outra linguagem vai ser moleza.

a) Download e Instalação

Instalar o Android Studio é compilar a sua primeira aplicação, é a parte mais chata e trabalhosa do processo.

Os pacotes são grandes e podem demorar para serem baixados e instalados, por isso seja muito paciente na hora de fazer este processo.

Também sugiro que você tenha um bom computador para estar utilizando o programa.

Trabalho com um processador i5 e 8gb de memória RAM, sugiro um computador equivalente ou próximo à isso.

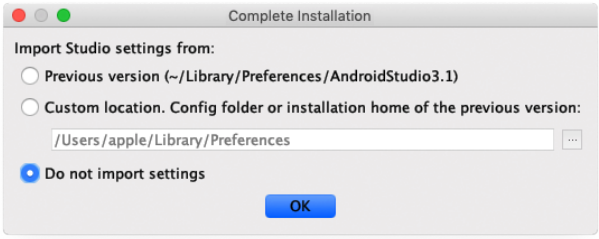
Não é necessário possuir um dispositivo Android pois existe a possibilidade de desenvolver e testar os aplicativos no simulador.

O primeiro passo é **baixar** e **executar** o instalador do Android Studio IDE através do link

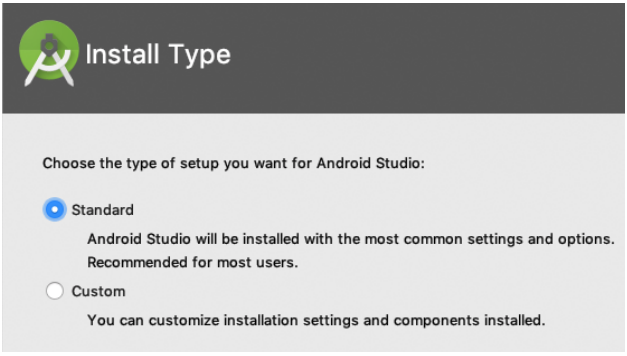
https://developer.android.com/studio/

Esta opção pede sua configuração de preferências, caso você já tenha instalado o Android Studio alguma vez anteriormente, pode aproveitar suas configurações anteriores.

Como é nossa primeira instalação ainda, deixe marcado em ***Do not import settings***.

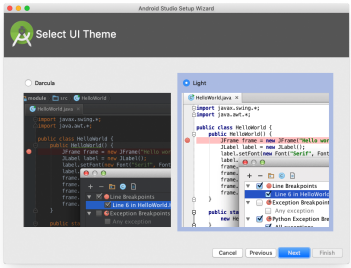


Prossiga com a instalação padrão (***Standard).***

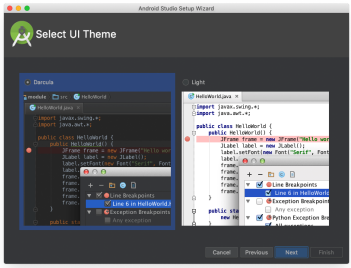


b) Temas e Preferências

Aqui você deve escolher entre um dos 2 **Temas de Cor** das janelas e menus do programa. Claro (Light) e escuro (Darcula).

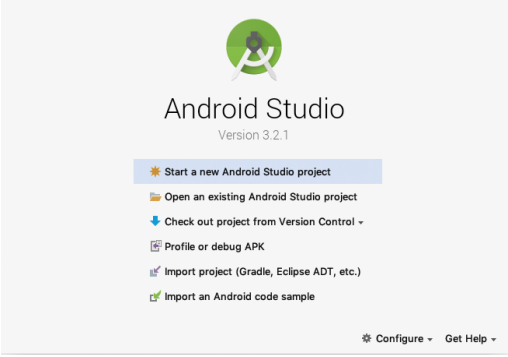


Light



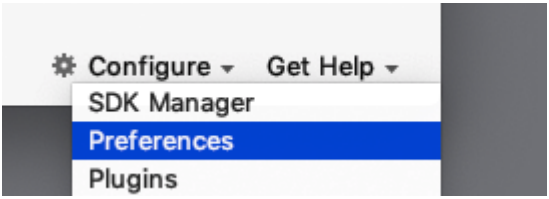
Darcula

Ao terminar a instalação, aparecerá a seguinte tela:



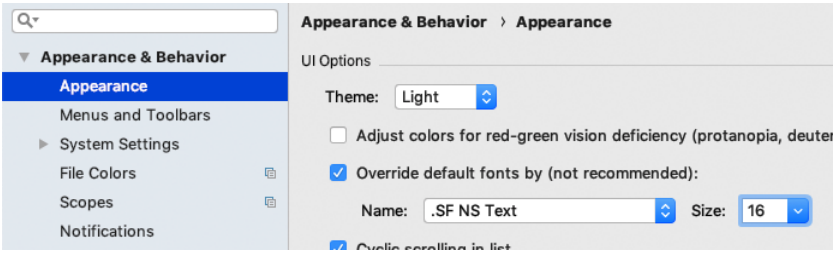
Antes de criar nosso primeiro projeto, vamos aprender como configurar o **tamanho dos textos** no programa.

Acesse a opção *Configure - Preferences*



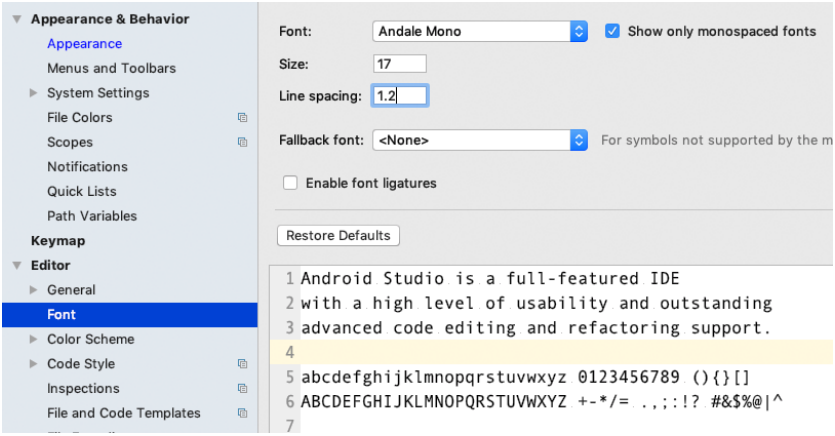
Em *Appearance - UI Option* pode-se alterar o tamanho dos textos da **interface do programa**.

Ligue a opção **Override default fonts** e deixe em o **Size** em **16**.



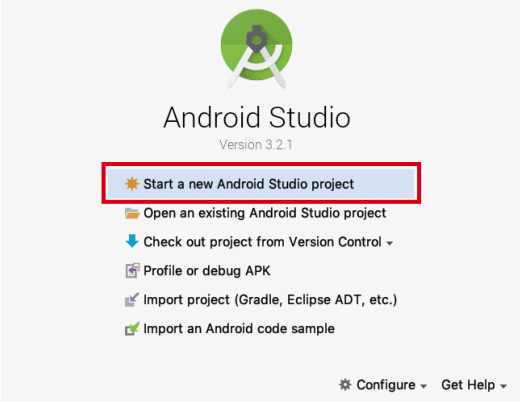
** Aumentar o tamanho do texto da interface, pode fazer com que alguns textos fique fora da margem em algumas janelas do programa, por isso existe a mensagem de não recomendado (not recommended). Farei apenas com a finalidade de expor melhor os textos e nomes neste material.*

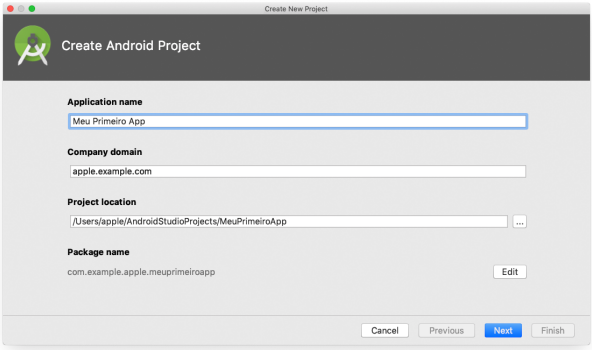
Nas opções Editor - Font podemos alterar o tamanho da fonte dos **códigos do projeto**. Deixe o **Size** em **17**.



c) Criando um Novo Projeto

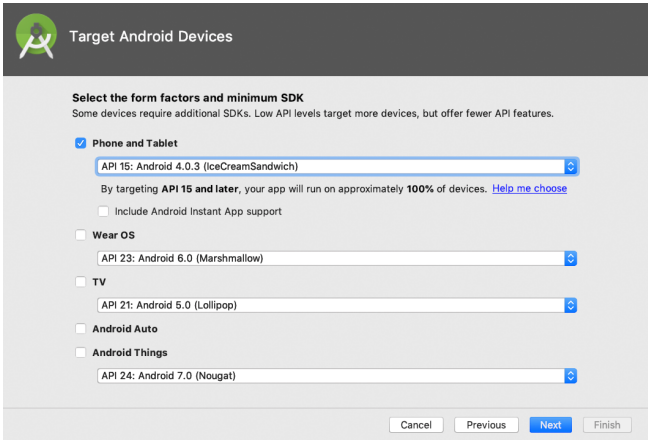
Agora sim crie seu primeiro projeto Android clicando na opção ***Start a new Android Studio project***.





Dê um nome do seu Aplicativo. Usarei o nome **Meu Primeiro App**.

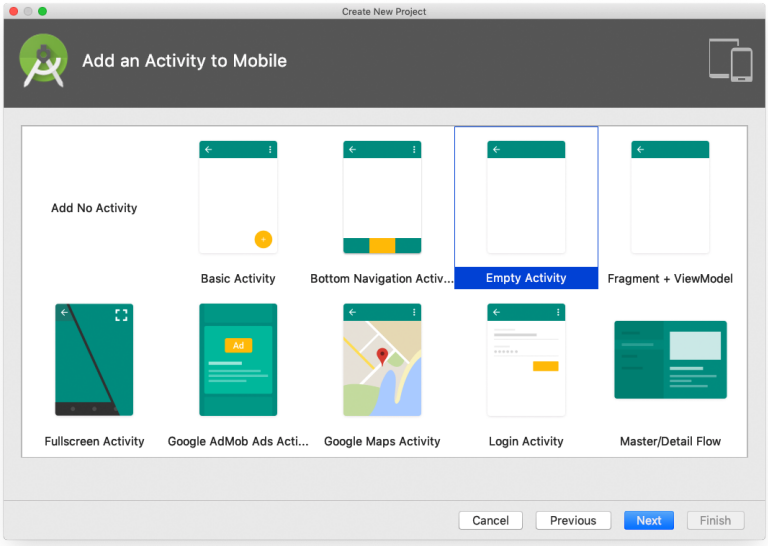
Agora selecione a versão mínima requerida para executar o seu aplicativo. Vou usar **API 16: IceCream Sandwich**



Isto garante que aparelhos com versões mais antigas do sistema Android consigam executar a sua aplicação.

O próximo passo é selecionar um template inicial para sua aplicação.

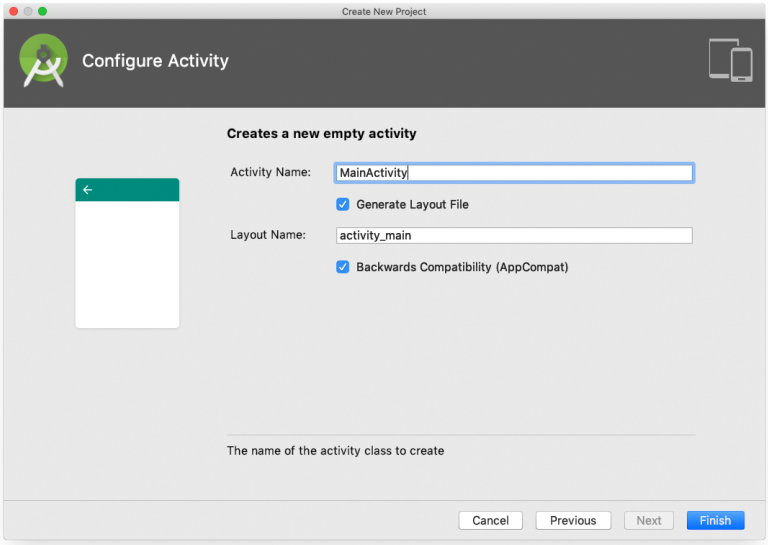
Utilize o modelo em branco **Empty Activity**.



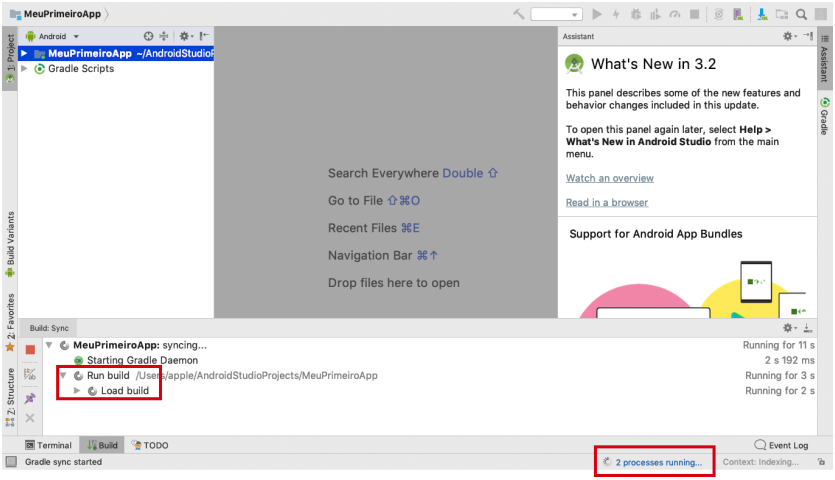
O próximo passo é definir o nome da classe da tela inicial da aplicação.

Este nome vem por padrão como MainActivity, deixarei este nome mesmo. O outro campo Layout Name serve

para definir o arquivo xml que vai montar o layout da nossa tela. Deixarei o nome padrão **activity_main**.



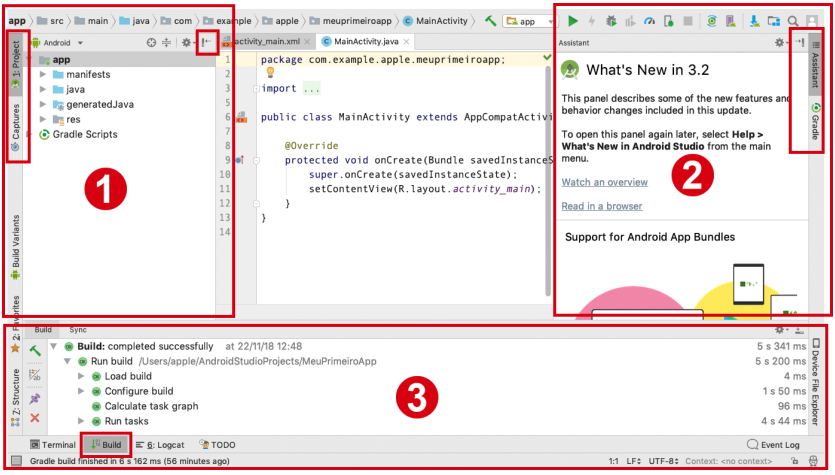
Siga com a criação do projeto até o final, e aguarde até que todos os arquivos e pastas do projeto sejam criados. Isto pode demorar até alguns minutos.



Observe as indicações de carregamento que irão aparecer na tela e espere todos os processos terminarem.

Quando o projeto acabar de ser criado e carregado, você vai enxergar uma tela assim.

Podemos dividir a interface do Android Studio em basicamente 3 áreas:



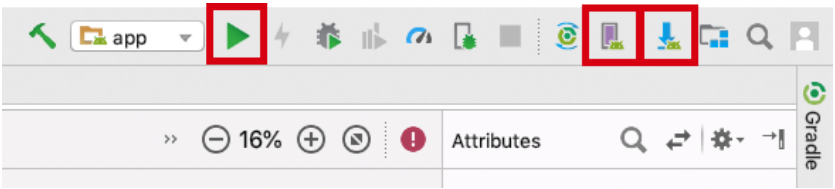
1) **Project**, onde você vai ter um explorador para navegar entre os arquivos que formam o projeto;

2) **Atributos** ou **Propriedades** dos objetos selecionados;

3) **Console** de Debug.

Clique nos seguintes ícones para expandir ou recolher cada uma das respectivas abas.

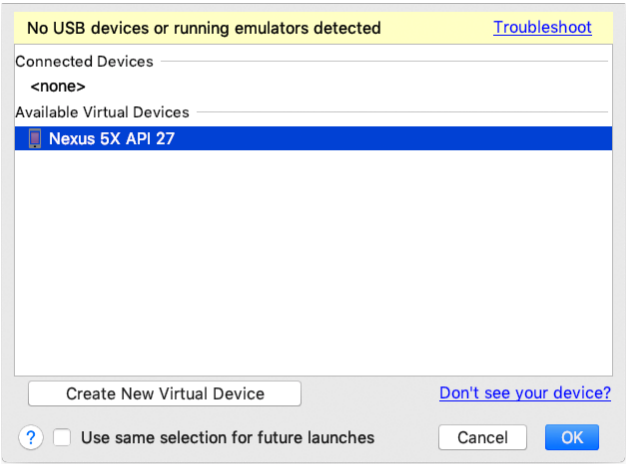
Preste atenção também nos seguintes comandos na barra superior: **Run AVD** e **SDK Manager**



d) Executando a Aplicação

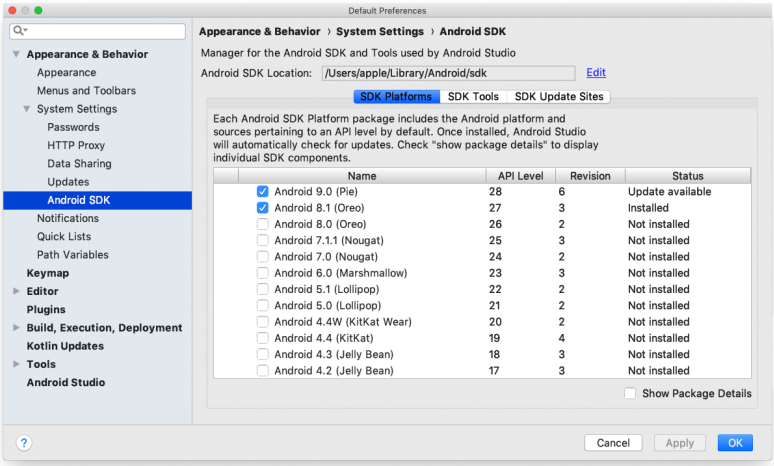
Para executar a aplicação clique no ícone do Play / Run.

Aqui vai aparecer a sua lista de simuladores criados (caso tenha algum) e a lista de aparelhos Android ligados na USB do seu computador. Selecione por onde deseja executar a aplicação.



e) Android SDK Manager

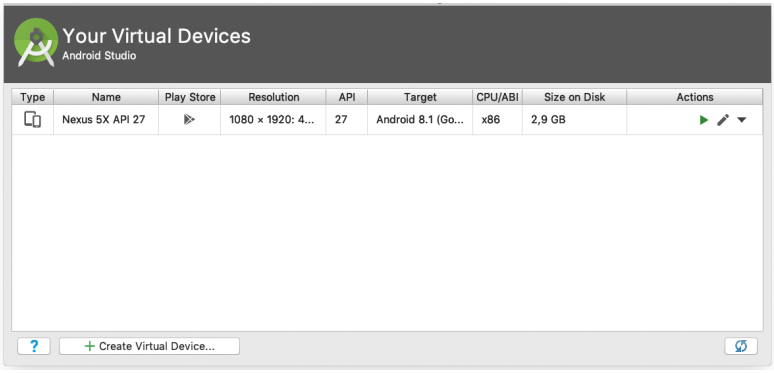
SDK Manager esta é a janela por onde controlamos o download dos pacotes e versões do android studio.



f) Simulador (Android Virtual Device)

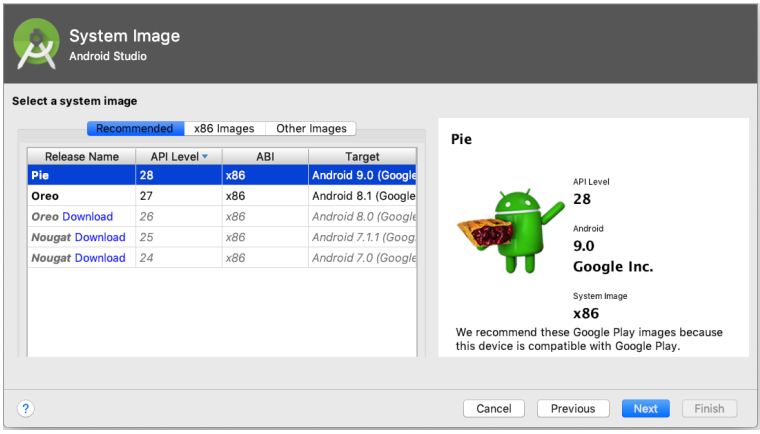
Android Virtual Device é o simulador virtual.

Se você não possui um dispositivo android, pode criar e utilizar um simulador para rodar a sua aplicação.



Ao abrir o AVD você verá a lista de simuladores criados.

Se ainda não existir nenhum, clique em **Create Virtual Device** e crie o simulador da última versão do android (Oreo ou Pie).



Caso não tenha essas versões instaladas, faça o download.

g) Android Studio e Simulador AVD - Resolução de Problemas

1) Faça todas as atualizações do android studio
2) Baixe o ultimo SDK e versão do AVD(OREO)

3) Tenha certeza que o intel HAXM esteja instalado
https://software.intel.com/en-us/articles/intel-hardware-accelerated-execution-manager-intel-haxm

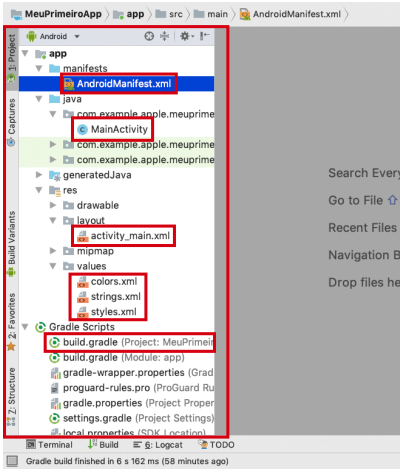
4) abra o Android Studio (algumas vezes) e deixe ele carregando todos os processos
(observe na barra inferior que ele fica fazendo download de pacotes ou atualizando sempre algo)

5) A Configuração recomendada para rodar o Android Studio e o Simulador é um i5 com 8gb de RAM (ou equivalente)

* Caso possua uma configuração inferior, recomendo executar os aplicativos no seu próprio aparelho Android, isso vai economizar bastante memória se não utilizar o simulador.

g) Anatomia do Projeto Android

Um projeto de android é composto por centenas de arquivos. Precisaremos entender apenas alguns deles para começarmos a mexer no nosso aplicativo.



Navegue na aba project (3) e encontre os seguintes arquivos:

- AndroidManifest.xml
- MainActivity
- Activity_main.xml
- Colors / Strings.xml
- build.gradle

Cada um deles é responsável por alguma tarefa ou configuração do nosso app.

Por enquanto, apenas lembre-se que o seu **código** principal da aplicação é a **MainActivity**. É nesta classe que vamos começar a programação do nosso app.

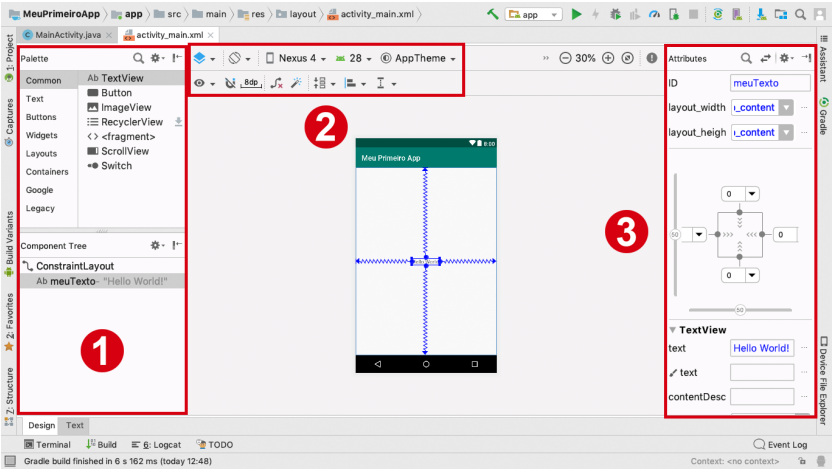
Lembre-se tambeém do **activity_main** que define o *layout* da nossa primeira tela.

h) Componentes Nativos

Acessando o **activity_main.xml** você vai para o editor de interfaces. Aqui podemos criar e configurar os **componentes nativos do Android** como janelas, botões,

textos, imagens ou qualquer outro componente visual de interface.

Você pode dividir esta tela em 3 áreas:



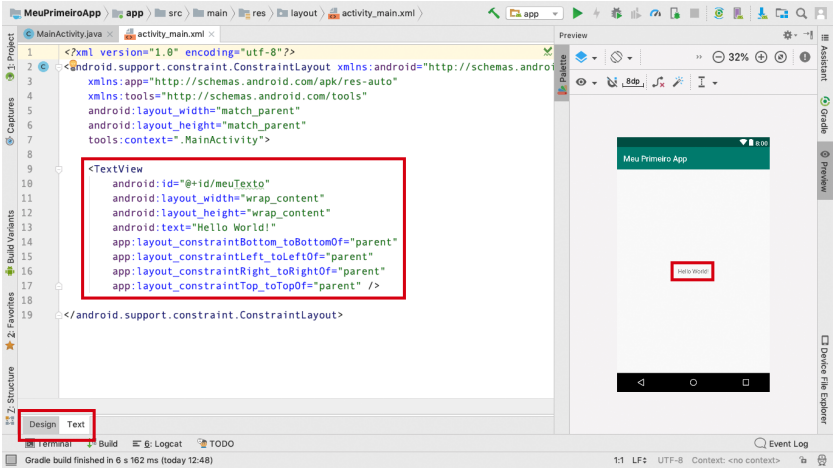
1. Paleta de Componentes e Hierarquia da View (Tela)
2. Ferramentas de Visualização e Layout
3. Propriedades dos Componentes

Nos botões abaixo, **Design** e **Text** podemos alternar entre **Layout** e **Código** .xml.

Perceba que toda a montagem no nosso layout é baseada em uma estrutura XML. Cada componente (botão, texto ou imagem) é definido através de *tags* que são montadas neste arquivo de texto enquanto estamos trabalhando com o editor.

Geralmente utilizamos o **Editor** para montar o nosso *layout* porém é interessante você observar e entender também como funcionam as *tags* do XML

.



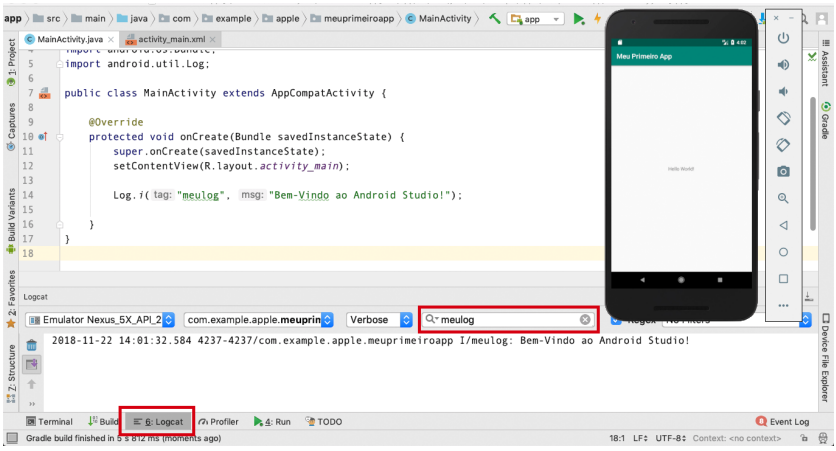
f) Hello World

Vamos agora escrever nosso primeiro trecho de código em Java dentro de uma aplicação Android.

Acesse o arquivo **MainActivity.java**. Esta é a classe que controla a primeira tela do nosso app.

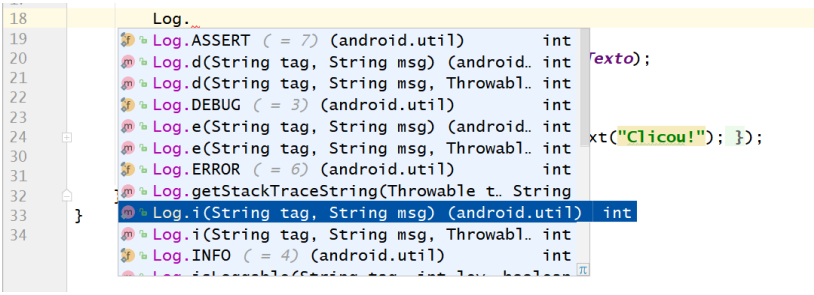
Perceba que ela estende a classe **AppCompatActivity**, então herda todas suas propriedades e métodos.

Insira um comando Log dentro da função on create.
O Log equivale ao comando System.out.print() que utilizamos anteriormente no compilador online.



Perceba que ao começar a digitar **Log**, uma janela vai se abrir com algumas sugestões de comando.

Selecione o **Log.i** e aperte o **Enter**.



Perceba que o comando já vai ser completado!

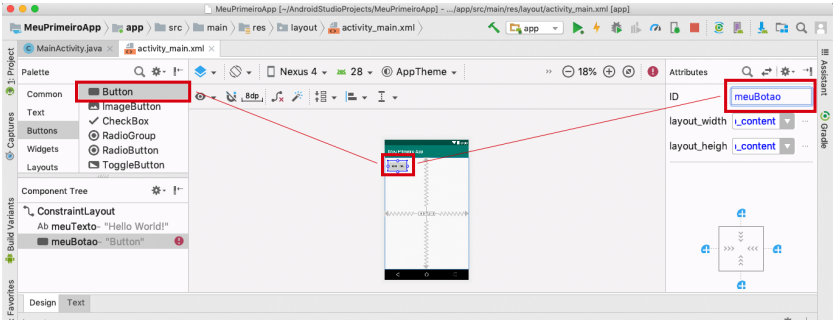
```
Log.i(tag:meulog, msg:"Bem Vindo ao Android Studio");
```

E se você observar logo no início do código, a biblioteca do Log foi importada também. (*import android.util.Log;*)

O Console do Android Studio exibe MUITAS mensagens de status e execução. Para isso existe a **tag**, ela vai nos ajudar a filtrar as mensagens do console e exibir apenas mensagens assinadas com a nossa **tag**.

Para isso, entre com o mesmo nome que você usou em **tag**, no campo de busca (ícone da lupa) nas opções do Console.

i) Identificando os Componentes



Cada componente no nosso *layout* possui uma identificação (ID).

Selecione algum componente como um botão por exemplo, e observe a aba de **atributos** do editor de layouts. Insira algum nome para identificá-lo.
Cada ID que associamos a um componente, fica registrado em um arquivo na **raíz do projeto do Android**, (R.id. …).

Usaremos este identificador posteriormente para manipular este componente no nosso código em Java.

Para o Android Studio, o componente visual ***Button***, não passa de um objeto criado a partir de uma classe, no caso a própria classe ***Button***.

Agora vamos para o arquivo ***MainActivity.java***, é lá onde vamos inserir nosso código em Java para começar a programar a interatividade do aplicativo.

Primeiro, precisamos criar um novo objeto na programação, este objeto vai ser do tipo ***Button***. É como declarar uma variável, do tipo ***Button***. Utilizamos:

```
Button botao = findViewById(R.id.meuBotao);
```

Onde ***Button*** é a classe/tipo do objeto, e ***botao*** é o nome da minha variável.

O comando **findViewById** vai procurar no layout, o objeto que tem o id associado.

j) Evento de Clique no Botão

Agora que já identificamos o botão no nosso código, vamos criar um evento de **Clique no Botão.**

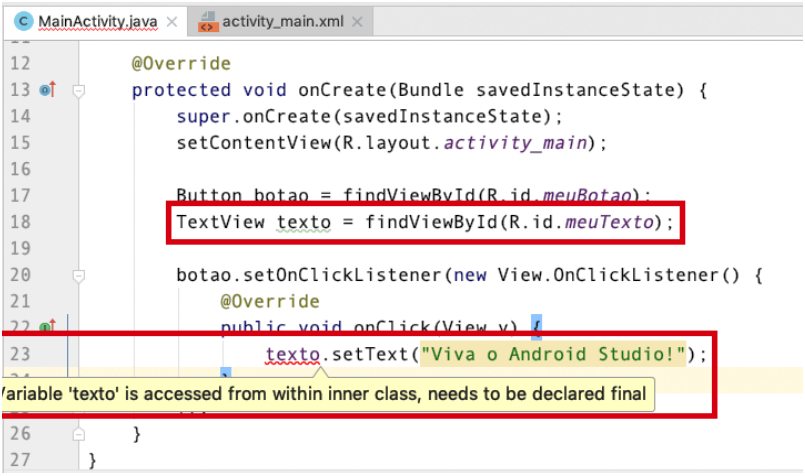
Digite cuidadosamente o código acima, perceba que o **TAB** autocompleta as expressões para você.

Dentro do método ***OnClickListener*** você pode inserir um **Log** apenas para testar.

A mensagem do ***Log*** deve ser exibida agora cada vez que clicarmos no botão.

k) Mudando o texto do TextView

Agora vamos fazer uma interação simples. Crie um componente de texto ***TextView*** no seu layout, identifique-o e declare no código, exatamente da mesma forma como fizemos com o componente ***Button***.



...

Parabéns querido aluno por chegar até aqui e ter adquirido mais este valioso conhecimento!

Se quiser aprender sempre mais sobre criação de Jogos e Aplicativos, não deixe de conhecer o Aprenda Programar, meu portal de cursos online onde você pode se especializar em:

- Algoritmos e Lógica de Programação
- Modelagem e Animação 3D
- Criação de Personagens para Jogos e Filmes
- Programação de Aplicativos Nativos para iOS e Android
- Criação de Games 2D, 3D e Realidade Virtual
- Realidade Aumentada e Visão Computacional
- Metodologia STEAM
- Robótica e Impressão 3D

Para virar aluno do Aprenda Programar você deve se inscrever pela plataforma Hotmart, no link abaixo.

Adquira seu acesso para sempre ao Aprenda Programar:
https://hotmart.com/product/en/aprenda-programar-com-tito-petri